IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Gordon J. Harris	Art Unit	: 2141
Serial No.	: 09/891,020	Examiner	: Quang N. Nguyen, Ph.D.
Filed	: June 25, 2001	Conf. No.	: 9419
Title	: TRUE ZERO-COPY SYSTEM AND METHOD		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

The brief fee of $500 is being filed concurrently through the EFS Electronic Filing System. Please apply all charges or credits to Deposit Account No. 06-1050, referencing attorney docket number 07072-137001.

(1) Real Party in Interest

The real party in interest is EMC Corporation.

(2) Related Appeals and Interferences

Appellant is not aware of any related appeal or interferences related to the above-identified patent application.

(3) Status of Claims

Claims 1-11, 14-20, and 23-25 are pending in this case. Claims 12-13 and 21-22 were canceled. The claims have been twice rejected. All pending claims are the subject of this appeal.

(4) Status of Amendments

All amendments have been entered. Appellant filed a Notice of Appeal on July 14, 2006.

(5) Summary of Claimed Subject Matter

Background

This invention relates to managing data movement between computers, and more particularly to an improved method and system of moving data between a network and a storage resource using a true zero-copy system and method.

Claim 1

Claim 1 is directed to a method that includes moving data from a network layer into a physical memory page. "For Ethernet, each physical page 52a-52n can be subdivided into cluster pairs (52a1-52a2), (52n1-52n2) which can be used to store network data packets 58a-58n that may arrive over the network 14." [Specification, FIG. 3A and page 9, lines 15-18]. "The memory system 36 also can store programs for facilitating the operation of the server 16 such as a network stack layer 40 which can include a transport control protocol/internet protocol

(TCP/IP) protocol stack for communicating over the network 14." [Specification, page 6, lines 18-22].

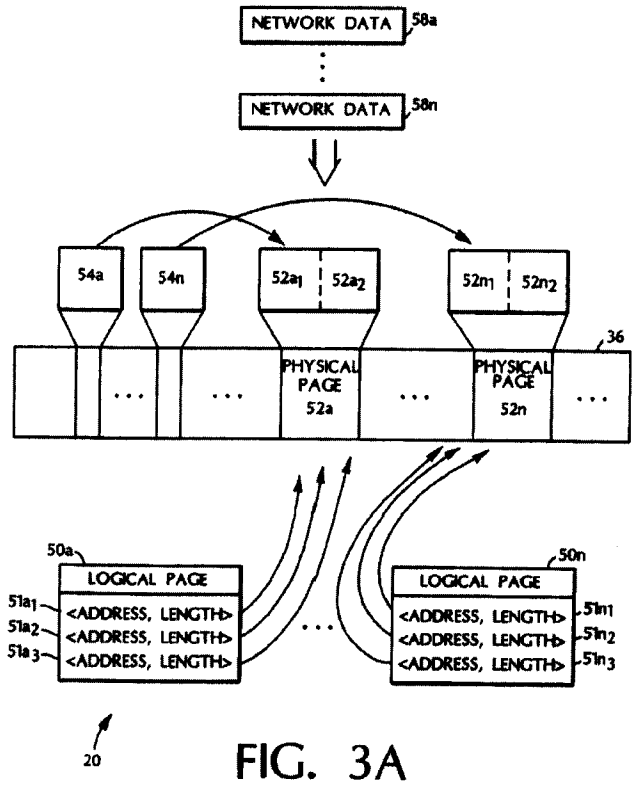

FIG. 3A

Claim 1 also recites that the network layer receives and transmits the data as data packets that are odd-sized, arrive asynchronously, and contain metadata embedded with real data. "For example, in an Ethernet network, network data can be transmitted and received as data packets that can be characterized as being odd-sized, arriving asynchronously or without warning, and having metadata such as protocol information embedded along with real data." [Specification, page 1, lines 8-12]. Claim 1 further recites that the physical memory page comprises a plurality of physical memory clusters. "Each physical page can be further subdivided into equally sized clusters." [Specification, page 9, lines 7-8].

Inventive features of claim 1 further include creating a logical page providing an aligned view of the data. "...the system 16 can create logical pages 50a and 50b. [Specification, page

15, lines 5-6]. "As a result, the logical page may need to include four physical page segments to represent 4KB of aligned data in the system." [Specification, page 21, lines 21-23].

Other inventive features of claim 1 include establishing a relationship between the logical page and the physical memory page such that the logical page is associated with said the plurality of physical memory clusters. "The logical pages 50a and 50b include references 51an and 51bn that point to the physical pages 52a, 52b, and 52c." [Specification, page 15, lines 6-8]. "The logical pages 50a and 50b are each associated with three clusters in physical memory having data totaling 4KB in length." [Specification, page 15, lines 20-22].

Further inventive features of claim 1 include forwarding a list of the logical pages to a storage resource such that the data referenced by the logical pages are stored subsequently into a storage resource. "The logical pages including the <address, length> pairs are passed to the disk controller device driver 46a. The <address, length> pairs can be used to reference the physical pages and to build a scatter/gather list in the DMA engine 23b of the disk controller 23a. The list can be used to write the data referenced by the logical pages to the corresponding disk blocks of file 29." [Specification, page 16, lines 1-7].

Claim 6

Claim 6 is directed to a computer system. "FIG. 2 is a system diagram illustrating a computer system 30 implementing a zero-copy method and system according to the present invention." [Specification, FIG. 2 and page 6, lines 3-5].

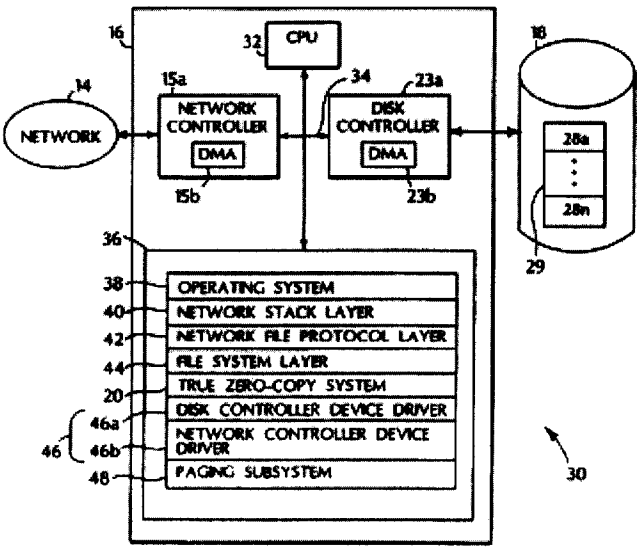

FIG. 2

The computer system of claim 6 includes a processor and a memory including at least one physical memory page and at least one logical page. "The server 16 can include a central processing unit (CPU) 32 such as an Intel Pentium connected to a memory system 36 over a computer bus 34." [Specification, page 6, lines 5-7].

The computer system of claim 6 further includes a network layer for receiving non-aligned data. "The memory system 36 also can store programs for facilitating the operation of the server 16 such as a network stack layer 40 which can include a transport control protocol/internet protocol (TCP/IP) protocol stack for communicating over the network 14." [Specification, page 6, lines 18-22].

Claim 6 recites that the network layer receives and transmits the data as data packets that are odd-sized, arrive asynchronously, and contain metadata embedded with real data. "For example, in an Ethernet network, network data can be transmitted and received as data packets that can be characterized as being odd-sized, arriving asynchronously or without warning, and having metadata such as protocol information embedded along with real data."

Other inventive features of claim 6 include a storage resource providing aligned data. "Disk data blocks 28 stored on the data storage device may be characterized as fixed in length and aligned in the data storage device 18 allowing it to be easily managed." [Specification, page 5, lines 15-17].

Claim 6 recites that the processor is configured to move data from a network layer into a physical memory page, the physical memory page comprising a plurality of physical memory clusters. "For Ethernet, each physical page 52a-52n can be subdivided into cluster pairs (52a1-52a2), (52n1-52n2) which can be used to store network data packets 58a-58n that may arrive over the network 14." [Specification, page 9, lines 15-18].

Claim 6 recites that the processor is also configured to create a logical page providing an aligned view of the data. "...the system 16 can create logical pages 50a and 50b. [Specification, page 15, lines 5-6]. "As a result, the logical page may need to include four

physical page segments to represent 4KB of aligned data in the system." [Specification, page 21, lines 21-23].

Claim 6 recites that the processor is further configured to establish a relationship between the logical page and the physical memory page such that the logical page is associated with the plurality of physical memory clusters. "The logical pages 50a and 50b include references 51an and 51bn that point to the physical pages 52a, 52b, and 52c." [Specification, page 15, lines 6-8]. "The logical pages 50a and 50b are each associated with three clusters in physical memory having data totaling 4KB in length." [Specification, page 15, lines 20-22].

Claim 6 recites that the processor is even further configured to forward a list of the logical pages to a storage resource such that the data referenced by the logical pages are stored subsequently into a storage resource. "The logical pages including the <address, length> pairs are passed to the disk controller device driver 46a. The <address, length> pairs can be used to reference the physical pages and to build a scatter/gather list in the DMA engine 23b of the disk controller 23a. The list can be used to write the data referenced by the logical pages to the corresponding disk blocks of file 29." [Specification, page 16, lines 1-7].

Claim 14

Claim 14 recites an article comprising a computer-readable medium that stores computer executable instructions. "In a third aspect, the invention provides an article comprising a computer-readable medium that stores computer executable instructions for controlling the operation of a computer." [Specification, page 3, lines 7-10].

Inventive features of claim 14 include instructions for causing the computer to move data from a network layer into a physical memory page. "For Ethernet, each physical page 52a-52n can be subdivided into cluster pairs (52a1-52a2), (52n1-52n2) which can be used to store network data packets 58a-58n that may arrive over the network 14." [Specification, page 9, lines 15-18]. "The memory system 36 also can store programs for facilitating the operation of the server 16 such as a network stack layer 40 which can include a transport control protocol/internet protocol (TCP/IP) protocol stack for communicating over the network 14." [Specification, page 6, lines 18-22].

Claim 14 also recites that the network layer receives and transmits the data as data packets that are odd-sized, arrive asynchronously, and contain metadata embedded with real data

and the physical memory page comprises a plurality of physical memory clusters. "For example, in an Ethernet network, network data can be transmitted and received as data packets that can be characterized as being odd-sized, arriving asynchronously or without warning, and having metadata such as protocol information embedded along with real data." [Specification, page 1, lines 8-12]. "Each physical page can be further subdivided into equally sized clusters." [Specification, page 9, lines 7-8].

Inventive features of claim 14 further include instructions for causing the computer to create a logical page providing an aligned view of the data. "...the system 16 can create logical pages 50a and 50b. [Specification, page 15, lines 5-6]. "As a result, the logical page may need to include four physical page segments to represent 4KB of aligned data in the system." [Specification, page 21, lines 21-23].

Other inventive features of claim 14 include instructions for causing the computer to establish a relationship between the logical page and the physical memory page such that the logical page is associated with the plurality of physical memory clusters. "The logical pages 50a and 50b include references 51an and 51bn that point to the physical pages 52a, 52b, and 52c." [Specification, page 15, lines 6-8]. "The logical pages 50a and 50b are each associated with three clusters in physical memory having data totaling 4KB in length." [Specification, page 15, lines 20-22].

Further inventive features of claim 14 include instructions for causing the computer to forward a list of the logical pages to a storage resource such that the data referenced by the logical pages are stored subsequently into a storage resource. "The logical pages including the <address, length> pairs are passed to the disk controller device driver 46a. The <address, length> pairs can be used to reference the physical pages and to build a scatter/gather list in the DMA engine 23b of the disk controller 23a. The list can be used to write the data referenced by the logical pages to the corresponding disk blocks of file 29." [Specification, page 16, lines 1-7].

(6) Grounds of Rejection

(1) Claims 1-11, and 14-18 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Nijhawan et al. (U.S. Patent 6,374,341) in view of Vishin et al. (U.S. Patent

5,860,146) and in further view of Applicant Admitted Prior Art (AAPA) described on page 1, lines 8-12 of the background of Appellant's specification.

(2) Claims 19-20 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Nijhawan, Vishin, AAPA, and further in view of Richter et al., (U.S. Publication 2003/0097481 A1).

(3) Claims 23-25 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Nijhawan, Vishin, AAPA, and further in view of Westbrook et al., (U.S. Patent 6,934,760).

(7) Argument

Obviousness

"It is well established that the burden is on the PTO to establish a prima facie showing of obviousness, *In re Fritsch*, 972 F.2d. 1260, 23 U.S.P.Q.2d 1780 (C.C.P.A., 1972)."

"It is well established that there must be some logical reason apparent from the evidence or record to justify combination or modification of references. *In re Regal*, 526 F.2d 1399 188, U.S.P.Q.2d 136 (C.C.P.A. 1975). In addition, even if all of the elements of claims are disclosed in various prior art references, the claimed invention taken as a whole cannot be said to be obvious without some reason given in the prior art why one of ordinary skill in the art would have been prompted to combine the teachings of the references to arrive at the claimed invention. Id. Even if the cited references show the various elements suggested by the Examiner in order to support a conclusion that it would have been obvious to combine the cited references, the references must either expressly or impliedly suggest the claimed combination or the Examiner must present a convincing line of reasoning as to why one skilled in the art would have found the claimed invention obvious in light of the teachings of the references. *Ex Parte Clapp*, 227 U.S.P.Q.2d 972, 973 (Board. Pat. App. & Inf. 985)."

"The mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification." *In re Gordon*, 221 U.S.P.Q. 1125, 1127 (Fed. Cir. 1984).

> Although the Commissioner suggests that [the structure in the primary prior art reference] could readily be modified to form the [claimed] structure, "[t]he mere fact that the prior art could be so modified would not have made the modification obvious unless the

> prior art suggested the desirability of the modification." *In re Laskowski*, 10 U.S.P.Q. 2d 1397, 1398 (Fed. Cir. 1989).

"The claimed invention must be considered as a whole, and the question is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination." *Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick*, 221 U.S.P.Q. 481, 488 (Fed. Cir. 1984).

> Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination. Under Section 103, teachings of references can be combined only if there is some suggestion or incentive to do so. *ACS Hospital Systems, Inc. v. Montefiore Hospital*, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984) (emphasis in original, footnotes omitted).

"The critical inquiry is whether 'there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination.'" *Fromson v. Advance Offset Plate, Inc.*, 225 U.S.P.Q. 26, 31 (Fed. Cir. 1985).

"The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. " *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990).

"Obviousness may not be established using hindsight." *Kahn v. Gen. Motors Corp.*, 135 F.3d 1472, 1479 (Fed.Cir.1998)

Discussion

1. Claims 1-11, and 14-18 are allowable over Nijhawan in view of Vishin in further view of AAPA.

Claims 1, 2, 4, 6, 7, 9, 10, 14, 15 and 17

For the purposes of this appeal only, claims 1, 2, 4, 6, 7, 9, 10, 14, 15, and 17 stand or fall together. Claim 1 is representative of this group of claims.

Appellant's claim 1 is directed to a method that includes moving data from a network layer into a physical memory page, wherein the network layer receives and transmits the data as data packets that are odd-sized, arrive asynchronously, and contain metadata embedded with real data and the physical memory page comprises a plurality of physical memory clusters.

The Examiner acknowledges that "Nijhawan does not explicitly teach moving data from a network layer into a physical page, wherein the network layer receives and transmits the data as data packets that are odd sized, arrive asynchronously, and contain metadata embedded with real data." [Final Action page 4]. Appellant notes that Nijhawan provides no dislosure of a network. The Examiner, however, relies on Vishin and AAPA to teach the features lacking in Nijhawan.

In particular, the Examiner relies on Vishin to teach a network layer that receives and transmits data as data packets and moving the data from the network layer into a physical page. Specifically, on pages 4 and 11 of the Final Action, the Examiner cites FIG. 1 of Vishin and the accompanying passage at col. 1, lines 12-24, which is reproduced below for reference:

> Referring to FIG. 1, there is shown a distributed computer system 100 that includes a plurality of processor clusters 102. Each cluster 102 will typically include one to four symmetric processors 104 that are coupled via a common bus 106 to a local memory store 108. One or more of the clusters may also include secondary memory 110 (non-volatile memory such as magnetic disk storage). Access to the local memory store 108 (and secondary memory 110, if present) shared by all the processors in the cluster 102 is governed by a memory controller 112 that also can send requests via a network 114 to pull in pages of data stored in the memory stores or secondary memory of other clusters 102 or other devices coupled to the network 114.

On page 4 of the Final Action, the Examiner contends that the foregoing passage of Vishin "should include the step of moving data from a network layer into the memory stores, i.e., into a physical memory space." Appellant disagrees.

Appellant submits that Vishin's network (referred hereinafter as "network 114'), as described in FIG. 1 and the foregoing passage, is simply a connection of multiple processor clusters that is neither disclosed nor suggested whatsoever to move data from a network layer into a physical memory page. A network layer, according to its ordinary English meaning, is a communications layer that routes packets of data from a source network address to a destination network address. Vishin's network 114 is not understood to be a network layer or to have a

network layer. A network layer is never mentioned in Vishin; and furthermore, Vishin's description of the network 114 does not contain features that would suggest that the network 114 uses a network layer. Such features could include, for example, the routing of data packets, the use of a TCP/IP protocol, or an Ethernet implementation of the network 114.

On page 11 of the Final Action, the Examiner also contends that the foregoing passage of Vishin, which is presented above, "should include the steps of transmitting and receiving data as data packets before storing data into the memory stores..." Appellant disagrees.

There is nothing in the foregoing passage or anywhere else in Vishin that discloses or suggests that Vishin's network 114 transmits and receives data as data packets. Although Vishin describes transmitting data over the network 114, Vishin is silent as to whether the data is transmitted as packets or in some other form, e.g., serial bits. Absent any disclosure in Vishin of the type of the network 114 or the format of data transmitted over the network 114, Vishin's network 114 cannot be construed to be a network that transmits and receives data as data packets.

According to MPEP 2131.02 "A genus does not always anticipate a claim to a species within the genus." Furthermore, as set forth by case law, some motivation to select the claimed species or subgenus must be taught by the prior art, e.g., *In re Deuel*, 51 F.3d at 1558-59, 34 USPQ2d at 1215 ("No particular one of these DNAs can be obvious unless there is something in the prior art to lead to the particular DNA and indicate that it should be prepared."); *In re Baird*, 16 F.3d at 382-83, 29 USPQ2d at 1552; Bell, 991 F.2d at 784, 26 USPQ2d at 1531 ("Absent anything in the cited prior art suggesting which of the 1036 possible sequences suggested by Rinderknecht corresponds to the IGF gene, the PTO has not met its burden of establishing that the prior art would have suggested the claimed sequences.").

In this case, the network disclosed in Vishin is analogous to a broad genus of a network that can include many different species of networks, some of which may use a network layer to transmit data as data packets and some of which may not. Since Vishin offers no description of the network 114, other than that provided above in the passage col. 1, lines 12-24, the network 114 of Vishin could just as well be a non-packet network that does not transmit and receive data as data packets. For example, the network 114 could be implemented as a serial bus, a parallel

bus; a circuit switched network (e.g., a public switched telephone network); a time-division multiplexed network; an optical network, or one or more other non-packet networks.

Although Vishin's network, in theory, *could* be modified to transmit and receive data as data packets, contrary to the Examiner's assertion, there is nothing in Vishin that discloses or suggests that the network *should* include the steps of transmitting and receiving data as data packets. Thus, Examiner's statements that the network of Vishin includes a network layer that transmits and receives data as data packets are erroneous.

Claim 1 also calls for receiving and transmitting the data as data packets that are odd-sized, arrive asynchronously, and contain metadata embedded with real data. The Examiner cites Applicant Admitted Prior Art (AAPA) found on page 1, lines 8-12 of the Appellant's specification as teaching these features of claim 1. [Final Action page 4]. The AAPA is reproduced below for reference:

> For example, in an Ethernet network, network data can be transmitted and received as data packets that can be characterized as being odd-sized, arriving asynchronously or without warning, and having metadata such as protocol information embedded along with real data.

The Examiner contends that "it would have been obvious to one having ordinary skill in the art ... to combine the teachings of Nijhawan, Vishin, and AAPA to include moving data from a network layer into a physical memory page, wherein the network layer receives and transmits data packets that are odd-sized, arrive asynchronously, and contain metadata embedded with real data since such methods were conventionally employed in the art to extend the address space to memory outside the cluster by using [a] virtual memory management subsystem to manage access to [a] remote physical address through the use of a (remote) page table and/or an auxiliary translation lookaside buffer." [Final Action page 5 and restated on page 11]. Appellant disagrees.

In AAPA, data packets that are odd-sized, arrive asynchronously, and contain metadata embedded with real data are disclosed only in relation to an Ethernet network. As known in the art, an Ethernet network is a particular type of network that uses an Ethernet communication protocol to transmit and receive data as data packets. As discussed above, Nijhawan lacks disclosure of a network altogether and Vishin neither discloses nor suggests that the network 114

transmits and receives data as data packets. There is nothing in Vishin that suggests that network 114 is anything more than a simple connection between processor clusters, which could, for example, be implemented by a serial connection that sends data one bit at a time, not as packets. Thus neither Nijhawan nor Vishin provide any suggestion as to why one skilled in the art would be motivated to modify their systems with AAPA to transmit and receive data packets, and in particular, to transmit and receive data packets that are odd-sized, arrive asynchronously, and contain metadata embedded with real data.

Claims 3, 8, and 16

For the purposes of this appeal only, claims 3, 8, and 16 stand or fall together. Claim 3 is representative of this group of claims.

Claim 3 adds the distinct feature of "creating a plurality of logical pages based on the offset and length of the data associated with a network write operation." FIG. 4 and the accompanying passage on pages 16 and 17of Appellant's specification provide further description of this feature:

> Upon receipt of the data packets 71a through 71f, the computer system 16 can process the file write request 72 including the <file-name, offset, length> information. For example, the system 16 determines that the file 29 corresponds to the "file-name" in the write request. File 29 is represented by disk blocks 28a through 28n. Each disk block represents a fixed amount of data, 4KB in this example. In addition, each disk block is associated with an "offset" representing the offset from the beginning of the file and a "length" representing the length of data in each disk block. For example, the file 29 contains disk data blocks 28a, 28b, through 28n.
>
> The system 16 needs to move the data packets that have been received and subsequently stored in the physical pages to the corresponding disk blocks 28a and 28b. As a result, the system 16 can create logical pages 50a and 50b. The logical pages 50a and 50b include references 51an and 51bn that point to the physical pages 52a, 52b, and 52c. As discussed above, the references can include an <address, length> pair where the address corresponds to the address of the cluster portion of the physical page and the "length" corresponds to the length of data in the cluster portion of the physical page.

The Examiner asserts that the passages of Nijhawan at column 7, lines 3-18, and at column 8, lines 40-51 teach the features of claim 3 [Final Action, page 5]. The first of these passages describes various examples in which it may be desirable to provide physical pages of

variable size, e.g., that are larger in size than a typical fixed size of 4K, and has nothing to do with creating logical pages. The second of these passages, which is reproduced for reference below, describes an example in which the offset of a physical page is used to generate a logical page number:

> For example, if the page being mapped by 32-bit linear address 81 is an 8K page, then the upper 9 bits of the 10+12 bit offset can be used to define the page number such that a 19-bit logical page number is provided and a 13-bit offset is provided as an offset within the 8K page. If the page being mapped by 32-bit linear address 81 is a 4 M page, then all of the upper 10 bits of the 10+12 bit offset can be used as an extension to the offset such that a 10-bit logical page number is provided and a 22-bit offset is provided as an offset within the 4 M page. Accordingly, pages of sizes between 4K and 4 M can be mapped with 4K granularity using 32-bit linear address 81.

Although the foregoing passage of Nijhawan describes using the offset of an 8K page to generate a logical page number, the 8K page is neither disclosed nor suggested to be data associated with a network write operation. Rather, according to the passage of Vishin at col. 3, lines 55-59, a page is a division of physical memory that stores data: "For example, in a paging system using only fixed 4 K (Kilobyte) pages (and assuming no segment extensions), the microprocessor considers the overall physical memory address space as being 4 G in size divided into 2^{20} pages of 4 K each." There is nothing in Nijhawan that discloses or suggests that a logical page number or an associated logic page is based on the offset and length of the data associated with a network write operation. Rather, as disclosed in the passage of Nijawan at col. 4, lines 3-6, a logical page is created based on a mapping to a physical page: "a logical page number is mapped or translated to a physical page number, and then the target physical location within the page can be accessed." Accordingly, in Nijhawan, a logical page can assist a network write operation with accessing data stored in a physical page that is mapped to the logical page. However, the logical page of Nijhawan, for example, is nowhere described to be created based on the offset and length of the data associated with the network write operation.

Claims 5, 11, and 18

For the purposes of this appeal only, claims 5, 11, and 18 stand or fall together. Claim 5 is representative of this group of claims.

Claim 5 adds the distinct feature of "merging an existing physical memory cluster with a new physical cluster based on the offset and length of the existing physical memory cluster and based on the offset and length of the new physical memory cluster."

The Examiner cites the passage of Nijhawan at col. 9, line 61 to col. 10, line 7 as teaching the features of claim 5. The passage is reproduced below for reference:

> For example, when allocating memory from Microsoft Windows.TM., the required alignment can be guaranteed by allocating twice as much needed, locking the correct amount down in Microsoft Windows.TM., mapping it, and freeing the rest of the allocated memory if possible. For example, to map a 40K block of physical memory using a memory allocator of an OS such as Microsoft Windows.TM., 128K of memory can be allocated. Thus, within the allocated 128K of memory, the OS inherently maps a 64K block of memory that resides in physical memory on 64K boundaries, which can be used for the desired 40K block, and then the extra memory on either side of the 64K boundaries can be reallocated.

The foregoing passage of Nijhawan discloses allocating and reallocating blocks of physical memory depending on memory requirements of an operating system. The Examiner incorrectly interprets the term "reallocated" as meaning the same as "remerged" or "merged." [Final Action, page 6]. Allocating memory blocks is not the same a merging memory blocks. As described in Appellant's specification in FIG. 7A-D and on page 22, lines 11-22, the merging of an existing physical memory cluster with a new physical memory cluster involves combining the existing and new clusters such that they occupy either the same, overlapping, or contiguous segments of physical memory. Allocating and reallocating memory blocks, by contrast, simply involve assigning and reassigning blocks of memory without necessarily combing them. For example, FIG. 9 and accompanying passage at col. 10, lines 7-16, of Nijhawan shows allocating a block 92 of physical memory for storing a block A of memory, and similarly allocating a block of 94 of physical memory for storing a block B of memory. As blocks A and B are shown in FIG. 9 to occupy non-contiguous blocks 92 and 94, clearly, the allocation of blocks A and B to physical memory blocks 92 and 94 do not involve merging either of blocks A and B or blocks 92 and 94.

2. Claims 19 and 20 are allowable over Nijhawan in view of Vishin in

view of AAPA in futher view of Richter.

Claims 19 and 20

The Examiner acknowledges that "Nijhawan-Vishin-AAPA ... does not explicitly teach the network layer uses a transport control protocol / internet protocol (TCP/IP) to transmit and receive the data as data packets over a computer network such as an Ethernet." [Final Action, page 7]. However, the Examiner relies on Richter to teach this lacking feature. Furthermore, the Examiner asserts that

> it would have been obvious ... to combine the teaching of Nijhawan-Vishin-AAPA and Richter to use TCP/IP to transmit and receive the data as data packets over the Ethernet network since such methods were conventionally employed in the art to allow the system using the well-established networking protocol stack TCP-UDP/IP suite to verify end-to-end data integrity to ensure that intermediate forwarding notes, client memory problems, and statistically remote errors have not corrupted the original data packets outside of media layer detection in transmitting and receiving the data packets over the Ethernet network.

Richter is directed to a checksum method for detecting errors and verifying data in the form of data packets. For example, in paragraph [0003], part of which is reproduced below for reference, Richter states that a checksum method is performed on data packets and can be implemented at the network layer:

> Thus, as part of TCP-UDP/IP network protocol suite, checksum algorithms are implemented in order to verify data integrity of network packets that have traversed various network segments. Checksum algorithms have been implemented for the TCP-UDP layers (transport layers) and the IP layer (a network layer).

Assuming that the Examiner can combine Richter with Nijhawan, Vishin, and AAPA to arrive at the features of claims19-20, to establish a prima facie case of obviousness, the Examiner must show that the combination of references, when considered as a whole, suggest the desirability of their combination. The Appellant submits that the Examiner has not provided reasoning that meets the legal standard required to establish a prima facie showing of obviousness.

In this regard, there is nothing in Richter that describes or suggests the use of translation lookaside buffers such as those disclosed in Nijhawan or Vishin. Furthermore, as discussed above, neither Nijhawan nor Vishin disclose or suggest anything that would motivate a person of ordinary skill in the art to modify their systems to transmit data over a network as data packets, a feature required by AAPA and Richter. Therefore, to combine Nijhawan, Vishin, AAPA, and Richter in the manner suggested by the Examiner, a person of ordinary skill in the art would need to rely upon the Appellant's invention as a roadmap. The Examiner's proffered combination of Nijhawan, Vishin, AAPA, and Richter amounts to a hindsight combination, which is improper as a matter of law:

> Determination of obviousness must be based on consideration of the claimed invention "as a whole," and cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the patented invention; in making this "as a whole" assessment, court requires a showing that an artisan of ordinary skill in the art at the time of invention, confronted by the same problems as the inventor and with no knowledge of the claimed invention, would select the various elements from the prior art and combine them in the claimed manner. *Medtronic Xomed, Inc. v. Gyrus ENT LLC* (F.Supp.2d, 2006 WL 2147704, M.D.Fla.,2006).

Appellant submits that a person of ordinary skill in the art would not look to the teachings of AAPA and Richter when faced with the problems of Nijhawan and Vishin, and vice versa, as Nijhawan and Vishin are directed to virtual memory management using translation lookaside buffers and AAPA and Richter are directed to the transmission of data packets over an Ethernet network, which is non-analogous art.

3. Claims 23-25 are allowable over Nijhawan in view of Vishin in view of AAPA in futher view of Westbrook.

Claims 23-25

Claims 23-25 recite that "the data packets arrive in a sequence that is different from an original sequence in which they were transmitted." The Examiner acknowledges that "Nijhawan-Vishin-AAPA ... does not explicitly teach the data packets arrive in a sequence that is different from an original sequence in which they were transmitted." [Final Action, page 8].

However, the Examiner relies on Westbrook to teach this lacking feature. Furthermore, the Examiner asserts that it would have been obvious to a person of ordinary skill in the art to combine the teachings of Westbrook with Nijhawan, Vishin, and AAPA because "it is typically more cost effective and technically feasible to provide multiple slower rate links or switching paths, than to provide a single higher rate path and such designs also achieve other desired performance characteristics."

Assuming that the Examiner can combine Westbrook with Nijhawan, Vishin, and AAPA to arrive at the features of claims 23-25, to establish a prima facie case of obviousness, the Examiner must show that the combination of references, *when considered as a whole*, suggest the desirability of their combination. The Appellant submits that the Examiner has not met the burden required to establish a prima facie showing of obviousness.

In this regard, there is nothing in Westbrook that describes or suggests the use of translation lookaside buffers such as those disclosed in Nijhawan or Vishin. Furthermore, as discussed above, neither Nijhawan nor Vishin disclose or suggest anything that would motivate a person of ordinary skill in the art to modify their systems to transmit data over a network as data packets, a feature required by AAPA and Westbrook. Nijhawan and Vishin are clearly directed to subject matter that is non-analogous to the subject matter of AAPA and Westbrook.

Furthermore, the Examiner's proffered motivation for combining Westbrook with Nijhawan, Vishin, and AAPA describes advantages (i.e., "more cost effective and technically feasible multiple slower rate links or switching paths") that are not applicable to Nijhawan, Vishin, or AAPA. For example, nowhere does Nijhawan, Vishin, or AAPA describe or imply the use of multiple slower rate links or switching paths.

Nijhawan, Vishin, AAPA, and Westbrook, when considered as a whole, do not disclose or suggest the desirability of making their combination. Thus, to combine these references in the manner suggested by the Examiner, a person of ordinary skill in the art would need to use the Appellant's invention as a roadmap to select the references and piece them together. The Examiner's proposed combination of Nijhawan, Vishin, AAPA, and Westbrook amounts to a hindsight combination, which as discussed above, is improper as a matter of law.

Conclusion

Appellant submits that the examiner erred in rejecting Appellant's claims and should be reversed.

Respectfully submitted,

Date: September 6, 2006

Frank R. Occhiuti
Reg. No. 35,306

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21410080.doc

Appendix of Claims

1. A method comprising:

moving data from a network layer into a physical memory page, wherein the network layer receives and transmits the data as data packets that are odd-sized, arrive asynchronously, and contain metadata embedded with real data and the physical memory page comprises a plurality of physical memory clusters;

creating a logical page providing an aligned view of the data;

establishing a relationship between the logical page and the physical memory page such that the logical page is associated with the plurality of physical memory clusters; and

forwarding a list of the logical pages to a storage resource such that the data referenced by the logical pages are stored subsequently into a storage resource.

2. The method of claim 1 further comprising:

dividing the physical memory pages into physical memory clusters such that the data received by the network layer is stored into the physical memory clusters.

3. The method of claim 1 further comprising:

creating a plurality of logical pages based on the offset and length of the data associated with a network write operation.

4. The method of claim 1 further comprising:

creating a read only logical page comprising zeros.

5. The method of claim 1 further comprising:

merging an existing physical memory cluster with a new physical cluster based on the offset and length of the existing physical memory cluster and based on the offset and length of the new physical memory cluster.

6. A computer system comprising:

a memory including at least one physical memory page and at least one logical page;

a network layer for receiving non-aligned data, wherein the network layer receives and transmits the data as data packets that are odd-sized, arrive asynchronously, and contain metadata embedded with real data;

a storage resource providing aligned data; and

a processor configured to:

move data from a network layer into a physical memory page, the physical memory page comprising a plurality of physical memory clusters,

create a logical page providing an aligned view of the data,

establish a relationship between the logical page and the physical memory page such that the logical page is associated with the plurality of physical memory clusters, and

forward a list of the logical pages to a storage resource such that the data referenced by the logical pages are stored subsequently into a storage resource.

7. The system of claim 6 wherein the processor is further configured to divide the physical memory pages into a memory cluster such that the data received by the network layer is stored into the memory cluster.

8. The system of claim 6 wherein the processor is further configured to create a logical page layer based on the offset and length of the data associated with a network layer write operation.

9. The system of claim 6 wherein the processor is configured to create a read only logical page of zeros.

10. The system of claim 6 wherein the processor is configured to create a read only logical page of uninitialized data.

11. The system of claim 6 wherein the processor is further configured to merge an existing physical memory cluster with a new physical memory cluster based on the offset and length of the existing physical memory cluster and based on the offset and length of the new physical memory cluster.

12 and 13. (cancelled)

14. An article comprising a computer-readable medium that stores computer executable instructions for causing a computer to:

move data from a network layer into a physical memory page, wherein the network layer receives and transmits the data as data packets that: are odd-sized, arrive asynchronously, and

contain metadata embedded with real data and the physical memory page comprises a plurality of physical memory clusters;

create a logical page providing an aligned view of the data;

establish a relationship between the logical page and the physical memory page such that the logical page is associated with said the plurality of physical memory clusters; and

forward a list of the logical pages to a storage resource such that the data referenced by the logical pages are stored subsequently into a storage resource.

15. The article of claim 14 further including instructions to divide the physical memory pages into physical memory clusters such that the data received by the network layer is stored into the physical memory clusters.

16. The article of claim 14 further including instructions to create a logical page based on the offset and length of the data associated with a network write operation.

17. The article of claim 14 further including instructions to create a read only logical page of zeros.

18. The article of claim 14 further including instructions to merge an existing physical memory cluster with a new physical cluster based on the offset and length of the existing physical memory cluster and based on the offset and length of the new physical memory cluster.

19. The method of claim 1 wherein the network layer uses a transport control protocol / internet protocol (TCP/IP) to transmit and receive the data over a computer network.

20. The method of claim 19 wherein the computer network is an Ethernet.

21 and 22. (cancelled)

23. The method of claim 1, wherein the data packets arrive in a sequence that is different from an original sequence in which they were transmitted.

24. The method of claim 6, wherein the data packets arrive in a sequence that is different from an original sequence in which they were transmitted.

25. The method of claim 14, wherein the data packets arrive in a sequence that is different from an original sequence in which they were transmitted.

Evidence Appendix

None.

Related Proceedings Appendix

None.